Записки научных
семинаров ПОМИ
Том 258, 1999 г.

S. A. Klioner

SYMBOLIC COMPUTATION WITH INDEXED OBJECTS WITHIN *Mathematica*

ABSTRACT. A new *Mathematica* package EinS is intended for calculations involving sums of indexed objects (e.g., tensors). EinS automatically handles implicit summations and dummy indices, allows one to assign symmetry properties to new objects and has an efficient built-in simplification algorithm based on pattern matching. Further features include printing expressions in a 2-dimensional form, exporting into plain TEX or LATEX with user-controllable alignment commands, converting implicit summations into explicit ones, debugging capabilities and online help messages. Some typical applications of EinS are described.

1. Introduction

General Theory of Relativity is one of the traditional fields of application of computer algebra. Complexity of calculations typical for virtually any serious problem in this area makes manual computation extremely tedious and unreliable. Starting from the 1960s many computer algebra systems were developed for the use specifically in General Relativity and more generally in the area of metric theories of gravitation [15, 4]. It is necessary to distinguish between indicial tensor computation and computation of components of tensors. These two classes of packages are quite different. The latter usually allows one to compute a set of quantities for given components of metric: components of Christoffel symbols, components of curvature tensor, etc. Examples of such packages are SHEEP (including CLASSI), EXCALC and REDTEN for REDUCE, CTENSR for Macsyma, and recently developed GRTensor/II for MAPLE, and TTC, CARTAN and GRTensorM for *Mathematica* and a stand-alone package ORTHOCARTAN. This paper is devoted mostly to indicial tensor calculations. Here, indices (of tensors or other objects) remain symbols and software operates with symbolic relations between various indexed objects. The principal aim of a package for indicial tensor computation is to handle implicit (Einsteinian) summation rules and to be able to simplify expressions involving dummy indices and user-defined symmetries of various objects. Among packages for indicial tensor computation one can mention ITENSR for Macsyma and STENSOR for SHEEP, both appeared in the late 1970s, and newer packages MathTensor and RICCI for *Mathematica* and ATENSOR for REDUCE.

However, each particular field of application of metric gravity theories has its own specific language (i.e., a set of notations, conventions, etc.) and put forward specific requirements to the computer algebra software. The author of this paper works in the field of astronomical applications of metric gravity theories. Typical calculations in this field represent operations with power series (e.g., in powers of c^{-1} as within the post-Newtonian approximation scheme) involving many indexed objects (functions, tensors, etc.) with various symmetry properties as well as partial derivatives of the objects with respect to coordinates or other parameters. Implicit (Einstein) summation notation is widely used here. Another important property is that usually "time" and "spatial" values of indices are treated separately, and in some cases covariant and contravariant "spatial" indices are not distinguished at all. A look at the existing packages for indicial tensor calculation showed that none of the existing systems allowed us to handle automatically all the properties of our research field mentioned above.

This forced us to implement our own package for operations with indexed objects (not necessarily tensors). The package is called EinS which stands for "Einstein Summation handler". EinS is a package for *Mathematica* which is one of the best modern computer algebra systems. This allows one to use the full power of *Mathematica* to treat the problem under study. For example, such facilities of *Mathematica* as arbitrary precision arithmetic, simplification of rational expressions, symbolic differentiation and integration, highly controllable substitution mechanism, flexible formatted output system, etc. are very important for applications. EinS is a relatively small package. It consists of about 3000 lines of *Mathematica* top level code including on-line help messages. EinS is a flexible package which is relatively easy to alter to solve any reasonable problem involving indexed objects. General design and functionality of the package resulted from scientific problems in the field of astronomical applications of metric gravity theories, main requirements of which are listed above. EinS automatically handles implicit summations and dummy indices, allows one to assign symmetry properties to new objects and has an efficient built-in simplification algorithm based on pattern matching. Further features include printing expressions in a 2-dimensional form, exporting into plain TEX or LATEX with user-controllable alignment commands, converting implicit summations into explicit ones, debugging capabilities and online help messages. EinS supports also two kinds of symbolic indices: "time-space" ones (say, running from 0 to 3 in standard notations of General Relativity), and purely "spatial" ones (running from 1 to 3).

EinS (as well as *Mathematica* itself) is not intended to perform calculations with very large number of terms. The most cumbersome computation which has been done with the use of EinS was computing the Landau-Lifshits pseudotensor for a generic metric containing terms of order of $\mathcal{O}(c^{-1})$ in g_{0i}. The calculations involved about 2000 terms in the result and up to 5000 in intermediate expressions. Each term was a product of up to 10 indexed objects, some of which were defined to be symmetric. Current processors allow EinS to operate with 10000 or more terms depending on their structure. However, if one has to operate with much longer expressions (say, with $\sim$ 100000 terms) STENSOR or FORM [4] are probably a better choice.

Although there exist several packages for indicial tensor calculation, algorithms of indicial tensor operations are tricky, and every new attempt to implement them is of interest. That is why even a small summary published about EinS [7] attracted certain attention and a reference on EinS appeared in a recent review on applications of computer algebra in general relativity [4].

2. General structure of EinS

EinS works under *Mathematica* starting from the version 2.1. This allows one to employ the power of one of the most comprehensive computer algebra systems, but, on the other hand, it also requires from the user some knowledge of the *Mathematica*'s interface and its top level language. Although the semantics of the examples below are obvious, we suppose that the reader possesses certain knowledge of *Mathematica*. We describe below EinS by dividing its functions into several groups and giving examples for each group. In the examples below, lines starting with the symbol ">" are input from users and those starting with "<" are responses from EinS.

- Commands which represent several built-in objects. E.g., `LeviCivita[i,j,k]` and `Delta[i,j]` stand for the Levi-Civita and Kronecker symbols, respectively.
- Commands to define new indexed objects, their symmetries and their independence of certain parameters (for example, coordinates or time). E.g., the command

```
DefObject[ A, 4, 3,4, 1,2, 3,4 ]
> DefObject:: Object defined as follows:
<              input name: A, valence 4
<              symmetric in indices: 3,4
<              antisymmetric in indices: 1,2
<              indices number(s) 1,2 are printed
<               as superscripts
<              indices number(s) 3,4 are printed
<               as subscripts
```

defines an object `A` with 4 indices. `A` is antisymmetric with respect to its first two indices and symmetric with respect to the last two ones. The first two indices will be printed as superscripts while the last two ones as subscripts:

```
> A[i,j,k,l]
<  ij
< A
<    kl
```

When defining an object the user can specify the number of indexes of the object, its print name to be used instead of the input name, symmetry properties of the objects and, finally if each particular index of the object is to be printed as sub- or superscript. EinS is not specifically intended for tensors and does not distinguish automatically covariant and contravariant indices. However, its flexible printing capabilities allow one to mimic rigorous tensor notations if it is necessary.
Another example is the command

```
> DeclareConstant[gamma,0]
< DeclareConstant:: gamma, valence 0 is declared
<  constant
```

which defines the object `gamma` without indices to be constant. Under *Mathematica* 3.0 notebook interface EinS makes use of its new printing capabilities (e.g., by printing "space-time" dummy indices as greek letters, the Kronecker's `delta` as δ, etc.). The user can also specify any special characters as names of objects. However, due to many bugs and inconsistencies in the new *Mathematica's* output forms (`TraditionalForm`, `StandardForm`), it is still preferable to use terminal-like `OutputForm` to output expressions with many subscripts and superscripts.

- Commands to declare particular symbols to represent coordinates of a reference system. The command

```
> DefRS[x,t]
<                                    -1  0  i
< DefRS:: reference system (t = c   x ,x ) defined.
```

defines `x[i]` and `t` to represent coordinates and coordinate time $t = x^0/c$ of a reference system. This information is used primarily to output various expression in traditional notations. E.g., $\frac{\partial(\gamma A^{ij}{}_{kl})}{\partial x^s}$ can be defined as

```
> PD[gamma A[i,j,k,l],x[s]]
<         ij
< gamma A
<           kl,s
```

It is automatically simplified (taking into account that `gamma` is declared to be constant above) and printed in traditional notations.

- Commands to define relations between objects including implicit summation rules. E.g.,

```
> S = DefES[ A[i,j,k,l] Delta[i,j], i,j ];
```

defines `S` to represent $A^{ij}{}_{kl}\,\delta_{ij}$. EinS allows one to declare each dummy index appearing in an implicit summation to be "space-time" (running from 0 to N, where in default case of 4-dimensional space-time $N = 3$) or purely "spatial" (running from 1 to N). The latter case is default. EinS automatically distinguishes between these two kinds of indices. This allows one to use the package effectively when working with "3+1" split of space-time.

- Commands to simplify expressions. E.g.,

```
> ComputeES[S]
< 0
```

EinS simplifies the expressions into several steps:

(1) transforming each term into a simple "canonical" form (equal terms may still have different forms after this operation, but it decreases substantially the number of terms);

(2) subsequent *pattern matching* among the rest of terms with the full account for symmetries of the objects and the possibility to rename dummy indices;

(3) simplification of the built-in objects by using their pre-defined properties (e.g., $\delta_{ij}\,\delta_{jk} = \delta_{ik}$);

(4) checking if each term of the result can be further simplified in virtue of some additional circumstances (e.g., $A^{ij}\,B^{ij}$ will be simplified to zero if A is symmetric and B is antisymmetric as in the example above).

The steps can be executed manually one by one or automatically in a reasonable sequence (this can be done with the command `ComputeES` used above).

- Commands to print expressions in natural 2-dimensional form with "pretty" automatically generated (and controllable by the user) names for dummy indices

```
> S = ComputeES[
>       DefES[ A[i,j,k,s] Delta[j,k], j,k ]
>              ];
> PrintES[S]
<  ia
< A
<    sa
```

- Commands to convert implicit summations into partially (containing only "spatial" dummy indices) or fully explicit (containing no dummy indices) form. E.g., fully explicit form of `S` reads

```
> ToComponents[S]
<  i1      i2      i3
< A     + A     + A
<    s1      s2      s3
```

- Commands to export expressions into plain TEX or LATEX form with flexible automatically generated line breaking and alignment commands. This property is very important if one works with long expressions and is unique among all other indicial tensor packages. Here is an example of output with the LATEX alignment commands

```
> T = S + ( S /.  i->s, s->i  );
> PrintES[T]
<  sa      ia
< A     + A
<    ia      sa
> TeXDialect =LaTeX;
> $TermsPerLine=1;
> EinS2TeX[T]
< \begin{eqnarray}
< &&
< A^{sa\ \}_{\ \ ia}
< \nonumber \\
< &&
< +A^{ia\ \}_{\ \ sa}
```

```
< \nonumber
< \end{eqnarray}
```

This automatic output of EinS can be saved into a file and then edited with a plain text editor to get desired LATEX or TeX layout. Usually only a few changes are necessary (say, inserting `\label`, and deleting unnecessary `\nonumber`).

- A set of commands to save definitions of objects and expressions in a form which can be read back into EinS.
- A set of low level commands which can be used by experienced users to implement any kind of additional operations with indexed objects.
- Several built-in procedures simplifying debugging of the user programs.
- Online help messages for all commands.

All functions of EinS can work automatically with power series (say, in powers of c^{-1}) which allows one to use EinS effectively when working within typical approximation schemes.

3. Typical applications of EinS

Typical applications for which EinS has been actually written are various calculations in [the post-Newtonian approximation of] metric gravity theories. As an example, three typical applications will be outlined here.

3.1. "Canonical" form of the Landau-Lifshits pseudotensor.

The Landau-Lifshits pseudotensor is useful for a convenient formulation of the conservation laws in general relativity. One can find several forms of the Landau-Lifshits pseudotensor: that of Landau and Lifshits (actually two forms: through Christoffel symbols and through derivatives of $\sqrt{-g}\, g^{\alpha\beta}$, g being determinant of the metric tensor $g_{\alpha\beta}$; Section 101 of [14]) and that of Fock (Section 89 of [3]). Although these are various forms of the *same* quantity they look quite differently. In order to check that the forms coincide and to cast all of them into a form convenient from the computational point of view, a small program for EinS has been written, which produces a "canonical" form of the Landau-Lifshits pseudotensor $t^{\alpha\beta}$ starting subsequently from each of the forms mentioned above. The result can be written as the sum of the following 16 terms representing products (contractions) of the contravariant components of the metric $g^{\alpha\beta}$ and partial derivatives of the covariant components with respect to the coordinates $g_{\alpha\beta,\gamma}$:

$$
\begin{aligned}
t^{\alpha\beta} = \frac{c^4}{16\,\pi\,G}\Bigg(& - g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\chi\gamma\,,\iota}\, g_{\delta\varepsilon\,,\eta} \\
& + g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi} g^{\beta\delta}\, g_{\chi\eta\,,\iota}\, g_{\delta\varepsilon\,,\gamma} \\
& + g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\delta\gamma\,,\iota}\, g_{\varepsilon\eta\,,\chi} \\
& + g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\chi\gamma\,,\iota}\, g_{\varepsilon\eta\,,\delta} \\
& - g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\delta\eta\,,\iota}\, g_{\varepsilon\gamma\,,\chi} \\
& - g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\chi\eta\,,\iota}\, g_{\varepsilon\gamma\,,\delta} \\
& + g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\chi\delta\,,\eta}\, g_{\varepsilon\gamma\,,\iota} \\
& + \frac{1}{2}\, g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\varepsilon\iota\,,\delta}\, g_{\eta\gamma\,,\chi} \\
& - g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\varepsilon\eta\,,\delta}\, g_{\gamma\iota\,,\chi} \\
& - \frac{3}{2}\, g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\chi\delta\,,\eta}\, g_{\gamma\iota\,,\varepsilon} \\
& + \frac{1}{2}\, g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\chi\eta\,,\delta}\, g_{\gamma\iota\,,\varepsilon} \\
& + \frac{1}{2}\, g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\chi}\, g^{\beta\delta}\, g_{\delta\eta\,,\chi}\, g_{\gamma\iota\,,\varepsilon} \\
& - \frac{1}{4}\, g^{\chi\delta}\, g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\beta}\, g_{\chi\eta\,,\iota}\, g_{\delta\varepsilon\,,\gamma} \\
& + \frac{1}{2}\, g^{\chi\delta}\, g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\beta}\, g_{\chi\iota\,,\eta}\, g_{\delta\varepsilon\,,\gamma} \\
& - g^{\chi\delta}\, g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\beta}\, g_{\chi\gamma\,,\iota}\, g_{\varepsilon\eta\,,\delta} \\
& + \frac{3}{4}\, g^{\chi\delta}\, g^{\varepsilon\eta}\, g^{\gamma\iota}\, g^{\alpha\beta}\, g_{\varepsilon\eta\,,\delta}\, g_{\gamma\iota\,,\chi}\Bigg).
\end{aligned}
\qquad (1)
$$

Needless to say that the LATEX code for (1) has been produced by EinS. Note that the products of $g^{\alpha\beta}$ can be factored over the first 12 terms and over the last 4 terms in (llcanon), which makes it quite efficient for calculations. One more line

```
> ComputeES[ ll - ( ll /.  a -> b, b -> a  ) ]
< 0
```

allows one for check that (1) is symmetrical with respect to α and β. This fact is not obvious from (1) itself. Here, `ll` is a variable equal to the

Landau-Lifshits pseudotensor t^{ab} and indices a and b are equivalent to α and β in (1), respectively.

3.2. Landau-Lifshits pseudotensor for a generic metric.

For some calculations related to the post-Newtonian rotational equations of motion of an extended, arbitrary shaped body, it was interesting to compute the Landau-Lifshitz pseudotensor in a rigidly rotating reference system [8]. To this end, EinS has been used to compute the Landau-Lifshitz pseudotensor for the following generic metric

$$\begin{aligned} g_{00} &= 1 + \frac{1}{c^2}U + \frac{1}{c^4}W + \mathcal{O}(c^{-5}), \\ g_{0i} &= \frac{1}{c}R^i + \frac{1}{c^3}U^i + \frac{1}{c^5}W^i + \mathcal{O}(c^{-6}), \\ g_{ij} &= -\delta_{ij} + \frac{1}{c^2}U^{ij} + \frac{1}{c^4}W^{ij} + \mathcal{O}(c^{-5}). \end{aligned} \tag{2}$$

The computation involves substituting (2) into (1) and simplifying the result. Some properties of the result are presented in Table 1. It has been checked that for a "nonrotating" ($R^i = 0$) PPN reference system our results coincide with the known results for the Landau-Lifshits pseudotensor (see, e.g., Section 4.4 of [17]. The complete expressions for the pseudotensor for the metric (2) are available from the author.

Table 1: Some properties of the Landau-Lifshitz pseudotensor $t^{\alpha\beta}$ for the generic metric (2). Here g is the determinant of the metric tensor.

Component	Order of c	Depends on	No. of terms
$(-g)\,t^{00}$	c^2	R^i	3
	c^0	R^i, U, U^i, U^{ij}	27
	c^{-2}	$R^i, U, U^i, U^{ij}, W, W^i, W^{ij}$	167
$(-g)\,t^{0i}$	c^1	R^i, U, U^{ij}	16
	c^{-1}	$R^i, U, U^i, U^{ij}, W, W^{ij}$	157
$(-g)\,t^{ij}$	c^2	R^i	2
	c^0	R^i, U, U^i, U^{ij}	151
	c^{-2}	$R^i, U, U^i, U^{ij}, W, W^i, W^{ij}$	1268

3.3. Local reference systems in the PPN formalism.

Another interesting problem is a definition of the local reference system of a massive body in the PPN formalism. The problem can be described in the following way. From the physical point of view any reference system covering a region of space-time under study can be used to describe physical phenomena within that region. However, the reference systems which offer a simpler mathematical description of physical laws are more convenient for practical calculations. For a test observer one can construct a "proper" local reference system where gravitational field appears only as tidal terms and the coordinates in the infinitesimal vicinity of the observer are directly related to the observable quantities. For many applications related with physically adequate description of astronomical observations it is important to construct similar "proper" local reference system for a massive extended body (e.g., for the Earth). In general relativity this problem has been solved in the Brumberg-Kopeikin [1, 13, 12] and Damour-Soffel-Xu [DSX] formalisms. In both approaches a local reference system (T, X^i) of a massive extended body satisfies two conditions: (A) the gravitational field of external bodies is represented in the form of tidal potentials being $\mathcal{O}(\mathbf{X}^2)$; (B) the internal gravitational field of the body coincides with the gravitational field of a corresponding isolated source provided that the tidal influence of the external matter is neglected.

Generalization of the formalisms onto the framework of the Parametrized Post-Newtonian (PPN) formalism is very important, since modern astronomical observations are one of the most important sources of observational data for testing General Relativity and for estimating the values of the PPN parameters (e.g., β and γ). Therefore, in order to get physically meaningful estimates of the PPN parameters one must use physically adequate relativistic models of the observables. Many of these models require (e.g., the model for Very Long Baseline Interferometry (VLBI) which is the source of the best current estimate of γ) a rigorous definition of the local reference systems in the PPN formalism. The rigorous definition and its consequences are the subject of our several recent publications [10, 11]. For this work we made an extensive use of EinS. Mathematically the problem can be expressed as "matching" of the metric tensor $g_{\alpha\lambda}$ of the global PPN reference system (t, x^i), the metric tensor $G_{\mu\nu}$ of the local PPN reference system (T, X^i), and the coordinate transformations between them $X^\alpha(x^\lambda)$:

$$g_{\alpha\lambda}(t, \mathbf{x}) = \frac{\partial X^\mu}{\partial x^\alpha}\,\frac{\partial X^\nu}{\partial x^\lambda}\,G_{\mu\nu}(T, \mathbf{X}). \tag{3}$$

The metric tensor $g_{\alpha\lambda}$ of the global PPN reference system is known apriori. The local PPN metric tensor $G_{\mu\nu}$ as well as the coordinate transformations $X^\alpha(x^\lambda)$ are supposed to have some specific form containing a number of unknown functions to be derived with the aid of matching (3). EinS makes this task much more easy to complete: instead of several months of manual computation in General Relativity [12] only one week was required to write the corresponding program for EinS, debug it, and perform all necessary calculations and checks with the two PPN parameters β and γ [10, 11]. More detailed description of the use of EinS for this particular problem can be found in [9].

4. Possible future improvements of EinS

It is our intention to keep the package sufficiently small and easy to tune up for a particular problem. Potential users should not expect that EinS is too general. Furthermore, future developments of the package strongly depend on the scientific problems in which the author will be involved. Future improvements of EinS already scheduled in next releases of EinS are as follows.

- refining the simplification algorithm of EinS in three major directions:

(1) splitting of dummy indices into subgroups which cannot intersect apriori when performing pattern matching in the algorithm of simplification;

(2) handling more complicated symmetry properties including linear and possibly non-linear identities [16, 5, 6];

(3) automatic consistency check of symmetry properties (e.g., A^{ijk} should be immediately recognized to be zero if it is defined as $A^{ijk} = A^{jik}$ and $A^{ijk} = -A^{ikj}$ [4]).

- implementing of a set of procedures dealing with symmetric trace-free parts (STF) of the objects (computing the STF part of an arbitrary expression, taking into account the property of an object to be STF during simplification, etc.).

Further details on EinS as well as EinS itself are available from the author and the EinS's home page

`http://rcswww.urz.tu-dresden.de/~klioner/eins.html`

References

1. V. Brumberg , *Essential Relativistic Celestial Mechanics.* Adam Hilder, Bristol, 1991 .
2. T. Damour, M. Soffel, and C. Xu , *General relativistic celestial mechanics.* Physical Review D, **43** (1991), 3273–3307; Physical Review D, **45** (1992), 1017–1044; Physical Review D, **47** (1993), 3124 .

3. V. Fock , *The Theory of Space, Time and Gravitation.* 2nd revised ed., Pergamon Press, Oxford, 1964 .
4. D. Hartley , *Overview of computer algebra in relativity.* in: Relativity and Scientific Computing, F. Hehl, R. Puntigam, and H. Ruder, Eds. Springer, Berlin, 1996, pp. 173–191 .
5. V. Ilyin and A. Kryukov , *Symbolic simplification of tensor expressions using symmetries, dummy indices and identities.* in: ISSAC'91, Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation (Singapore, 1991), S. Watt, Ed., ACM Press, pp. 224–228 .
6. V. Ilyin and A. Kryukov , *A symbolic simplification algorithm for tensor expressions in computer algebra.* Programmirovanie (Computer Science) (January 1994), 83–91 .
7. S. Klioner , *EinS: A Mathematica package for tensorial calculation in astronomical applications of relativistic gravity theories.* in: Abstracts of GR14, the 14th international conference on general relativity (Turin, 1995), M. Francaviglia, Ed., SIGRAV-GR14, p. A.182 .
8. S. Klioner , *Angular velocity of extended bodies in general relativity.* in: Dynamics, ephemerides and astrometry in the solar system, S. Ferraz-Mello, B. Morando, and J. Arlot, Eds. Kluwer, Dordrecht, 1996, pp. 309–320 .
9. S. Klioner , *New system for indicial computation and its application in gravitational physics.* Computer Physics Communications (1998), accepted .
10. S. Klioner and M. Soffel , *Local reference systems with PPN parameters.* in: Proceedings of the 8th Marcell Grossmann Meeting (Singapore, 1998), R. Ruffini, Ed., World Scientific. in press .
11. S. Klioner and M. Soffel , *The Nordtvedt effect in rotational motion.* Physical Review D (1998), accepted .
12. S. Klioner and A. Voinov , *Relativistic theory of astronomical reference systems in closed form.* Physical Review D, **48** (1993), 1451–1461 .
13. S. Kopeikin , *Relativistic reference systems in solar system.* in: Itogi Nauki i Tekhniki, M. Sazhin, Ed., vol. 87. Nauka, Moscow, 1991, pp. 87–146 .
14. L. Landau and E. Lifshits , *The Classical Theory of Fields.* Pergamon Press, Oxford, 1971 .
15. M. MacCallum , *Symbolic computation in relativity theory.* in: EUROCAL'87, European Conference on Computer Algebra (Berlin, 1987), J. Davenport, Ed., Springer, pp. 34–43 .
16. A. Rodionov and A. Taranov , *Combinatorial aspects of simplification of algebraic expressions.* in: EUROCAL'87, Proceedings of the European Conference on Computer Algebra (Berlin, 1987), J. Davenport, Ed., Springer, pp. 192–201 .
17. C. Will , *Theory and experiment in gravitational physics, revised ed..* Cambridge University Press, Cambridge, 1993 .

Поступило 23 сентября 1999 г.

Lohrmann Observatory,
Dresden Technical University,
Mommsenstraße, 13, 01062 Dresden, Germany
klioner@rcs.urz.tu-dresden.de
http://rcswww.urz.tu-dresden.de/~klioner/